

MetaRule

Path Management

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2007 Cigital, Inc.

2007-03-29

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4593 bytes

Attack Category	<ul style="list-style-type: none">• Malicious Input	
Vulnerability Category	<ul style="list-style-type: none">• Buffer Overflow	
Software Context	<ul style="list-style-type: none">• File Path Management	
Location		
Description	<p>Output buffers for path functions must be sized to hold at least MAX_PATH characters.</p> <p>In general, Windows path functions need to have a buffer of at least MAX_PATH otherwise they are subject to buffer overruns.</p> <p>For some Windows functions, if a Unicode path begins with "\\?" then the path can be longer than MAX_PATH characters, up to 32,000+ characters in length. It is unclear when this applies in a way that would lead to buffer overflows.</p>	
APIs	Function Name	Comments
	PathAddBackslashA	
	PathAddBackslashW	
	PathAddExtension	
	PathAddExtensionA	
	PathAddExtensionW	
	PathAppend	
	PathAppendA	
	PathAppendW	
	PathCanonicalize	
	PathCanonicalizeA	
	PathCanonicalizeW	
	PathCombine	
	PathCombineA	
	PathCombineW	

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

Method of Attack			
Exception Criteria			
Solutions	Solution Applicability	Solution Description	Solution Efficacy
	Path functions.	Ensure that output buffer is sized as MAX_PATH characters.	Effective
Signature Details	Any function that returns a path as a result.		
Examples of Incorrect Code	<pre>WCHAR path[] = L"MyFile.dat"; // Buffer is too small LPWSTR pszPath = path; LPCWSTR dirs[] = { NULL }; if (!PathResolve(pszPath, dirs, PRF_VERIFYEXISTS) { handleError(); }</pre>		
Examples of Corrected Code	<pre>WCHAR path[MAX_PATH] = L"MyFile.dat"; // Buffer is correctly sized LPWSTR pszPath = path; LPCWSTR dirs[] = { NULL }; if (!PathResolve(pszPath, dirs, PRF_VERIFYEXISTS) { handleError(); }</pre>		
Source Reference	<ul style="list-style-type: none"> Rough Auditing Tool for Security (RATS)² 		
Recommended Resource			
Discriminant Set	Operating System	<ul style="list-style-type: none"> Windows 	
	Languages	<ul style="list-style-type: none"> C C++ 	

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com¹.

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. <mailto:copyright@cigital.com>